Study of AES implementation on FPGA by using Rijndael Algorithm

Lal Bahadur[1] Mr. Bhupendra Badoniya[2]
[1]M.Tech Scholar [2]Assistant Professor
Takshshila Institute of Engineering & Technology, Jabalpur (M.P), India.

Abstract— Cryptography was and still is one of the hot research areas. The growing demand for cryptography arises from the desire to secure networks and data against potential intruders. With the use of more handheld wireless devices and increasing networking and wireless data transfer, the issue of security is being addressed from many different directions. The National Institute of Standards and Technology (NIST) selected the Rijndael algorithm as a new Advanced Encryption Standard (AES) [1] in 2001. This standard was first developed for secure data encryption/ decryption for high-end applications. This paper presents a comprehensive survey on the use of Field programmable Gate Array (FPGA) in cryptographic applications as well as recent Research and Development (R&D) trends in this area. We discuss the advantages and drawbacks of FPGA for cryptographic applications. We then describe some countermeasures to overcome these drawbacks. Finally, we do a brief survey on Advanced Encryption Standard (AES) / Rijndael and Elliptic Curve Cryptography (ECC) algorithm implementations on FPGAs.

Keywords: cryptography, FPGA, AES, Rijndael, EEC, ASIC, SRAM

I. INTRODUCTION

The expanding use of digital communications, sensitive electronic financial transactions taking place over the Internet, and digital signature applications has emphasized the need for fast and secure communication networks to fulfil the requirements for secrecy, integrity, and non-reputability of exchanged information. Cryptography provides a method for securing and authenticating the transmission of information over insecure channels. Cryptography encodes the plaintext to a cipher text. There are two major classes of algorithms in cryptography: private-key algorithms and public-key algorithms. The most popular private-key algorithm is the Data Encryption Standard (DES), the first standardized crypto algorithm. Advanced Encryption standard (AES)/Rijndael on the other hand is the new encryption standard. The RSA is the most popular form of public-key algorithm. RSA stands for Rivest, Shamir, and Adleman, the inventors of the RSA crypto algorithm. Elliptic curve cryptosystems (ECCs) are cryptographic algorithms based on mathematical objects known as elliptic curves. Elliptic curve cryptography has been gaining in popularity recently. Lastly, the Diffie-Hellman key agreement protocol is a popular public-key technique for establishing secret keys over an insecure channel [2].

For performance as well as for physical security reasons, it is often advantageous to realize these cryptographic algorithms in hardware. Since modern security protocols are increasingly defined to be algorithm independent, a high degree of flexibility with respect to the cryptographic algorithms is desirable. A promising solution that combines high flexibility with the speed and physical security of traditional hardware is the implementation of cryptographic algorithms on reconfigurable devices such as FPGAs [3]. Implementation of security protocols on FPGA leads to the various advantages such as economical cost for low volume production, availability of sophisticated design and debugging tools, ability of in circuit reprogram ability and low start-up cost which leads to the lower financial risk in comparison to the fully customized Application Specific Integrated Circuits (ASIC's) and potentially much higher performance than software implementations.

II. FIELD PROGRAMMABLE GATE ARRAY

An FPGA is a logic chip comprising cells of logic blocks connected with software-configurable interconnections. Early FPGAs only contained small number of logic cells, and took entire seconds to reconfigure; today an FPGA contains hundreds of thousands of cells, and can be reconfigured in microseconds. It has the potential to provide the performance benefits of Application Specific Integrated Circuits (ASICs) and the flexibility of processors. Application specific hardware circuits can be created on demand to meet the computing and interconnect requirements of an application [4].

Current devices mainly use SRAM to control the configurations of the logic cells and the interconnection network. The configuration of a device can be modified by loading a bitstream onto its configuration memory. The size of a configuration bit-stream can range from several Kbits to several Mbits of data depending on the size of the corresponding FPGA.The evolution of the fabrication technology has led to the development of FPGAs with exceptional computational power at a lower cost. Existing state-of the- art FPGAs consists of two million gates and the next generation FPGAs will consist of approximately ten million gates. Finally, the cost, in terms of computational density as well as computational power, is rapidly decreasing making the FPGAs a cost-effective solution. Clearly, the density of FPGAs is growing faster than that of any general-purpose microprocessor. FPGA based solutions span from implementations that are occasionally upgraded in field to implementations that evolve during computation.

III. ADVANTAGES OF FPGA FOR CRYPTOGRAPHIC APPLICATIONS

The potential advantages of implementing cryptographic algorithms in FPGA are discussed briefly in this section. They include the following [5].

A. ALGORITHM ABILITY

It can be defined as the ability to switch between cryptographic algorithms during operation. Security protocols such as Secure Sockets Layer (SSL) and IP sec have been written to support multiple encryption algorithms. Whereas algorithm agility is costly with traditional hardware, FPGAs can be reprogrammed on the-fly.

B. ALGORITHM UPLOAD

It is perceivable that fielded devices are upgraded with a new encryption algorithm. Algorithm upload can be necessary because a current algorithm expired (e.g. DES), a new standard was created (e.g. Advanced Encryption Standard -AES). FPGA-equipped encryption devices connected to a network such as the Internet can upload the new configuration code. The upgrade of ASIC implemented algorithms is practically infeasible if many devices are affected or if the systems are not easily accessible.

C. THROUGHPUT

An ASIC encrypts at about double the speed of the FPGA. Although typically slower than an ASIC implementation, FPGA implementations run faster then software implementations.

D. ALGORITHM MODIFICATION

Algorithm modifications are easily made with FPGA. In many occasions cryptographic primitives or their modes of operation have to be modified according to the application.

E. ARCHITECTUTRE EFFICIENCY

In certain cases, hardware architecture can be much more efficient if it is designed for a specific set of parameters. These parameters can be the key, the underlying finite field, the coefficient used in an ECC system. FPGAs allow this type of design and optimization with specific parameter set.

F. COST EFFICIENCY

For low volume applications, implementation of a given algorithm in FPGA is much cheaper than an ASIC implementation. However, for high-volume applications ASIC solutions usually become the more cost-efficient choice

IV. DRAWBACKS OF FPGA FOR CRYPTOGRAPHIC APPLICATIONS

In this section we discuss the security problems produced by attacks against given FPGA implementations. These attacks are as follows [6]:

A. BLACK BOX ATTACK

This is the classical method to reverse engineer a chip. With this attack intruder is able to extract the inner logic of the FPGA.

B. READ BACK ATTACK

The idea of this attack is to read the configuration of the FPGA through the JTAG or the programming Interface in order to obtain the secret information.

C. CLONNING OF SRAM ON FPGAs

Normally, the data is stored externally in PROM and is transmitted to the FPGA at power-up in order to configure the FPGA. An intruder could easily tap the transmission and get the configuration file.

D. REVERSE ENGINEERING OF THE BITSTREAMS

The bit stream is not encrypted and hence by reverse engineering the bi stream an intruder can easily get the design of proprietary algorithms or the secret keys.

E. PHYSICAL ATTACK

This attack targets parts of the FPGA which are not available through the normal I/O pins. This can be achieved by using optical microscopes and mechanical probes. However, FPGAs are now becoming so complex that only advanced methods such as Focused Ion Beam (FIB) systems can help to launch such attack.

V. COUNTERMEASURES TO PREVENT THE ATTACKS

This section shortly summarizes possible countermeasures that can be provided to minimize the effects of the attacks mentioned in the previous section. Most of them have to be realized by design changes through the FPGA manufacturers, but some could be applied during the programming phase of the FPGA [6].

A. COUNTERMEASURE FOR BLACK BOX ATTACK

Due to the complex design of today's FPGAs, this attack is not a real threat nowadays. The complex nature of cryptographic algorithms such as AES, ECC and RSA also helps in preventing such attacks.

B. COUNTERMEASURE FOR READ BACK ATTACK

The read back attack can be prevented with the security bits set, as provided by the manufactures. If one wants to make sure that an attacker is not able to apply fault injection, the FPGA has to be embedded into a secure environment,

C. COUNTERMEASURE FOR CLONING OF SRAM FPGAs

One solution would be to check the serial number before executing the design and delete the circuit if it is not correct. Another solution would be to use dongles to protect the design. A more realistic solution would be to have the non-volatile memory and the FPGA in one chip or to combine both parts by covering them with epoxy. However, it has to be guaranteed that an attacker is not able to separate the parts. Encryption of the configuration file is the most effective and practical counter-measure against the cloning of SRAM FPGAs.

D. COUNTERMEASURE FOR PHYSICAL ATTACK

Counter techniques such as inserting dummy cycles into the circuit or applying an opposite current can be carried forward to FPGA applications.

E.*COUNTERMEASURE FOR SIDE CHANNEL ATTACK*

The methods can generally be divided into software and hardware countermeasures, with the majority of proposals dealing with software countermeasures. Software countermeasures refer primarily to algorithmic changes, such as masking of secret keys with random values, which are also applicable to implementations in custom hardware or FPGA. Hardware countermeasures deal either with some form of power trace smoothing or with transistor-level changes of the logic.

VI. FPGA IMPLEMENTATION OF CRYPTOGRAPHIC ALGORITHMS

In this section we will do a brief literature survey on the implementation of Advanced Encryption Standard / Rijndael and Elliptic Curve Cryptographic Algorithm on FPGAs. The performance of cryptographic implementation on FPGA is evaluated on the basis of throughput and the amount of hardware resources consumed to achieve this throughput.

A. ADVANCED ENCRYPTION STANDARD / RIJNDAEL ALGORITHM ON FPGA

Many implementations of Rijndael, the AES, have been presented using Xilinx FPGAs [7]. The implementation in [7] achieves a throughput of 6.956 Gbits/sec using a Xilinx Virtex-E. Implementing the Rijndael Byte-Sub operation using ROM resulted in a significant increase in throughput. Rijndael implementations achieved throughputs ranging from 570 Mbits/sec to 964 Mbits/sec depending on the implementation methodology using Xilinx . The implementation described in achieves a throughput rate of 17.8 Gbits/s for a 128-bit key through pipelining. The work presented in [8] presents a very compact implementation of AES on FPGA. It proposed a new way of implementing the Mix Columns and Inv Mix Columns transformations which reduces area. It achieves data streams of 150 Mbits/sec for encryption and decryption on a low cost Xilinx Spartan II FPGA using 222 slices and 3 BRAMs.

B. ELLIPTIC CURVE CRYPTOGRAPHY ON FPGA

In [9], a new elliptic curve crypto processor (ECP) architecture has been proposed for the computation of point multiplication for curves defined over Galois Field GF(p). The ECP has a scalable architecture in terms of area and speed specially suited for FPGAs. This processor uses a high-radix Montgomery multiplier that relies on the pre-computation of frequently used values and on the use of multiple processing engines. The ECP consists of main controller, arithmetic unit controller and the arithmetic unit. The authors developed a prototype that implemented ECP on a Xilinx FPGA. The ECP prototype used 11,416 LUTs, 5,735 Flip- Flops, and 35 Block RAMS. The frequency of operation of the prototype was 40 MHz for 192 bit operands and 37.3 MHz for the 521-bit multiplier.

VII. CONCLUSIONS

It is widely recognized that security issues will play a crucial role in many future computer and communication systems. A central tool for achieving system security are cryptographic algorithms. For performance as well as for physical security reasons, it is often required to realize cryptographic algorithms in hardware. Traditional ASIC solutions, however, have the well-known drawback of reduced flexibility compared to software solutions. A promising solution which combines high flexibility with the speed and physical security of traditional hardware is the implementation of cryptographic algorithms on reconfigurable devices such as FPGAs. This paper presented a comprehensive survey on the use of Field programmable Gate Array (FPGA) in cryptographic applications. We discussed the advantages and disadvantages of implementing cryptographic algorithms on FPGA. Countermeasures to overcome the various attacks were also discussed in this paper. Finally, the results of implementing Advanced Encryption Standard (AES) / Rijndael and Elliptic Curve Cryptography (ECC) algorithm implementations on FPGAs are discussed briefly.

REFERENCES

[1] National Institute of Standards and Technology: *FIPS 197: Advance Encryption Standard*, November 2001
[2] RSA Security Inc, "RSA Laboratories' Frequently Asked Questions About Today's Cryptography", Version 4.1,2000.
[3] T. Blum,and C. Paar, "High-radix Montgomery modular exponentiation on reconfigurable hardware," IEEE Transactions on Computers, vol. 50, No. 7, pp. 759- 764, July 2001.
[4] V. K. Prasanna, and A. Dandalis, "FPGA-based Cryptography for Internet Security," Online Symposium for Electronic Engineers, November 2000.
[5] A. Elbirt, W. Yip, B. Chetwynd, and C. Paar. An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists," IEEE Transactions on VLSI Systems, vol. 9, No. 4, pp. 545–557, August 2001.
[6] T. Wollinger, and C. Paar, " How Secure Are FPGAs in Cryptographic Applications?," in Proceedings of 13th International Conference on Field Programmable Logic and Applications - FPL 2003, Lisbon, Portugal, September 1-3, 2003.
[7] M. McLoone, and J. McCanny, "High Performance Single-Chip FPGA Rijndael Algorithm," in Workshop on Cryptographic Hardware and Embedded Systems CHES- 2001, C. K. Koc, D. Naccache, and C. Paar, Eds. Vol. LNCS 2162. Springer-Verlag, pp. 65-76, 2001.
[8] P. Chodowiec, and K. Gaj, "Very Compact FPGA Implementation of the AES Algorithm," in Workshop on Cryptographic Hardware and Embedded Systems CHES- 2003, C. Walter, C. K. Koc, and C. Paar, Eds. Vol. LNCS 2779. Springer-Verlag, pp. 319-333, 2003.
[9] G. Orlando, and C. Paar, "A Scalable GF(p) Elliptic Curve Processor Architecture for Programmable Hardware," in Workshop on Cryptographic Hardware and Embedded Systems CHES-2001, C. K. Koc, D. Naccache, and C. Paar, Eds. Vol. LNCS 2162. Springer-Verlag, pp. 348-363,2001.

